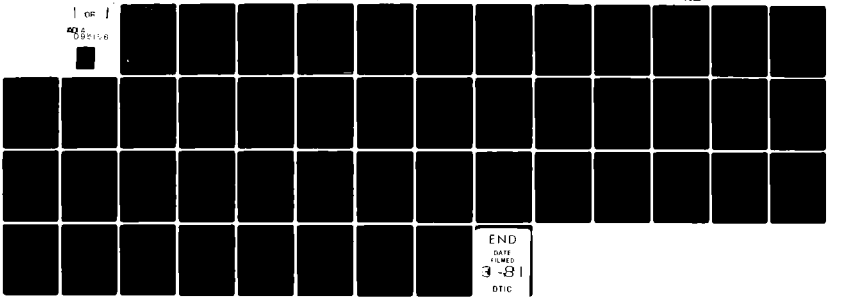
AD-A095 156 ARMY ENGINEER TOPOGRAPHIC LABS FORT BELVOIR VA F/G 8/2
TOPOGRAPHIC RELAXATION STUDY.(U)
SEP 79 M J HANNAH
UNCLASSIFIED ETL-0209 NL



ETL-0209

FINAL REPORT

Topographic Relaxation Study

Prepared for the
U. S. Army Engineer Topographic Laboratories
Fort Belvior, VA 22060

by

Marsha Jo Hannah
NASA/Ames Research Center
Institute for Advanced Computation

September 15, 1979

Approved for Public Release, Distribution Unlimited

Destroy this report when no longer needed.
Do not return it to the originator.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

The citation in this report of trade names of commercially available products does not constitute official endorsement or approval of the use of such products.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ETL-0209	2. GOVT ACCESSION NO. AD-A095156	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) TOPOGRAPHIC RELAXATION STUDY		5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Marsha Jo Hannah		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS NASA/AMES Research Center Institute for Advanced Computation P. O. Box 9071, Sunnyvale, CA 94086		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS R3205
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Engineer Topographic Laboratories Fort Belvoir, VA 22060		12. REPORT DATE 15 Sep 79 13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Topographic Relaxation
Slope Correction Algorithm
Terrain Classification Constraint on Digital Elevation Model

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Digital terrain models produced by computer correlation of stereo images are likely to contain occasional gross errors in terrain elevation. These errors typically result from having mismatched sub-areas of the two images, a problem which can occur for a variety of image-and terrain-related reasons. Such elevation errors produce undesirable effects when the models are further processed, and should be detected and corrected as early in the processing as possible.

SEE REVERSE SIDE

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

circ 3

IAC has developed algorithms to detect and correct errors in digital terrain models. These algorithms focus on the use of constraints on both the allowable slope and the allowable change in slope in local areas around each point. Relaxation-like techniques are employed in the iteration of the detection and correction phases to obtain best results.

PREFACE

This report describes the development effort performed by the Institute for Advanced Computation (IAC) for the U.S. Army Engineer Topographic Laboratories (ETL) under ETL Project No. 18R3205HT08.

The ETL project monitor for this effort was Mrs. Jane M. Brown of the Computer Sciences Laboratory (CSL) of ETL.

Table of Contents

List of Illustrations

SECTION 1

INTRODUCTION

Digital terrain models are playing an increasingly important role in many map-related tasks. These terrain models can be produced in several ways; in this research we have concentrated on elevation data resulting from digital correlation matching of sub-areas from stereo imagery [Crombie, 1976].

A significant problem in correlation-derived digital terrain models is the introduction of errors into the elevation data when the stereo matching algorithm produces mismatches. These mismatches can result from a variety of conditions, including low contrast in areas of the images, relief-induced distortions between the images, and the presence of ambiguities due to identical objects or highly periodic textures on the terrain. It is not yet feasible for a correlation algorithm to handle all of these difficult situations without error. For this reason, post-processing techniques have been sought to detect and correct errors which occurred in the correlation process. Work to date has centered on global techniques such as fitting polynomials to the data [Jancaitis, 1975] or filtering in either the spatial or frequency domains [Johnson, 1978].

Global techniques have the drawback that they give identical treatment to all areas of a digital terrain model. Terrain is rarely uniform in roughness, so uniform application of a global technique can produce oversmoothing in rough areas, while failing to correct errors in relatively flat areas. Local techniques, on the other hand, have the potential for coping with different terrain types within a model. Local techniques can also easily incorporate other terrain model information, such as land-use classifications.

In September, 1978, the Institute for Advanced Computation (IAC) was asked by the U.S. Army Engineer Topographic Laboratories (ETL) to explore local methods, suitable for parallel implementation, to detect and correct errors in digital terrain elevation data. The algorithms developed use constraints on the allowable slope and the allowable change-in-slope around each point in the terrain data. These measures are applied in parallel and iterated to achieve the desired results, in a manner analogous to the relaxation techniques employed by Rosenfeld [Rosenfeld, Hummel, and Zucker, 1976]. An investigation has also been made into the use of land-type classification data to influence the elevation correction process.

SECTION 2

THE BASIC ALGORITHMS

The primary objective of this research was the development of techniques for improving the internal consistency of digital elevation data. Given an array of elevation data H[I,J], the ideal result would be to modify each H[I,J] so that the slopes surrounding it become reasonable, consistent with one another, and consistent with neighboring slopes.

In this section, we present the basic algorithms suggested by ETL and discuss the results produced by these algorithms.

2.1 Mathematical Analysis

Ignoring boundary points for the moment, each elevation point H[I,J] has 8 neighbors, which can be described by their direction vectors [DI,DJ] relative to [I,J].

Direction	DI	DJ
1	0	1
2	1	1
3	1	0
4	1	-1
5	0	-1
6	-1	-1
7	-1	0
8	-1	1

The slope from the [I,J]-th point to its K-th neighbor is defined as

$$SLOPE[I,J,K] = (H[I+DI[K],J+DJ[K]] - H[I,J])/DIST[K] \qquad (2\text{-}1)$$

where DIST is the Euclidean base-plane distance between [I,J]'s grid point and that of the neighbor. ETL suggested that three sets of tests be performed on these slopes: the local neighbor slope consistency tests, the distant neighbor slope consistency tests, and the slope constraining tests.

The local neighbor slope consistency tests check the 4 pairs of slopes crossing a point to see that each pair is consistent, that is, that the slopes in each pair do not differ by more than a specified amount. If we define

$$DSLOPL[I,J,K] = SLOPE[I,J,K] - SLOPE[I-DI[K],J-DJ[K],K] \qquad (2\text{-}2a)$$

then this test requires that

$$ABS(DSLOPL[I,J,K]) \leq DSLTHRESH \qquad K=1:4 \qquad (2\text{-}2b)$$

The distant neighbor slope consistency tests check that the pairs of slopes approaching a point across each of the 8 neighbors are consistent. Here we define

$$DSLOPD[I,J,K] = SLOPE[I,J,K] - SLOPE[I+DI[K],J+DJ[K],K] \qquad (2\text{-}3a)$$

and require

$$ABS(\ DSLOPD[I,J,K]\ ) \leq DSLTHRESH \qquad K=1:8 \qquad (2\text{-}3b)$$

The slope constraining tests check each of the 8 slopes immediately surrounding a point to see that they are not unreasonable, i.e.

$$ABS(\ SLOPE[I,J,K]\ ) \leq SLTHRESH \qquad K=1:8 \qquad (2\text{-}4)$$

ETL also suggested that the elevation modification be accomplished by using the slope consistency tests as an indicator of which way and how much each elevation should be corrected. From Equations (2-1) and (2-2a) we see that DSLOPL[I,J,K] is positive when H[I,J] is less than the average of its two neighbors. Thus DSLOPL[I,J,K]>0 is an indication that H[I,J] should be moved upward, while DSLOPL[I,J,K]<0 indicates that it should be moved downward. DSLOPD[I,J,K] functions similarly.

ETL suggested thresholding, then combining these signed change-in-slope indicators, multiplied by a change-in-height variable, to develop the modification to each H[I,J]. If we define

$$TL[I,J,K] = \begin{cases} 1.0 & \text{if } DSLOPL[I,J,K] > DSLTHRESH \\ -1.0 & \text{if } DSLOPL[I,J,K] < -DSLTHRESH \\ 0.0 & \text{otherwise} \end{cases} \qquad (2\text{-}5a)$$

and

$$TD[I,J,K] = \begin{cases} 1.0 & \text{if } DSLOPD[I,J,K] > DSLTHRESH \\ -1.0 & \text{if } DSLOPD[I,J,K] < -DSLTHRESH \\ 0.0 & \text{otherwise} \end{cases} \qquad (2\text{-}5b)$$

then the suggested change in height at [I,J] is expressed as

$$DH[I,J] = DELHEIGHT*\left(\sum_{K=1}^{4} TL[I,J,K] + \sum_{K=1}^{8} TD[I,J,K] \right) \qquad (2\text{-}6a)$$

and the H[I,J] are modified by

$$H'[I,J] = H[I,J] + DH[I,J] \qquad (2\text{-}6b)$$

These steps were to be performed in parallel on all elements of the elevation data array, with the corrections being iterated until the data ceased to change meaningfully or went into an oscillatory condition.

2.2 Algorithm Results

This basic algorithm was implemented on IAC's TENEX system and applied to the data sets described in Appendix A. The results on the artificial data set, designated PATCH, illustrate some of the problems with this algorithm.

Consider the results on the following noise spikes for DSLTHRESH=.05 and DELHEIGHT=8/12. (For economy of space, we will present only one quadrant of the data; the results are, of course, symmetric.)

The data set

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
+8	0	0	0	0

converged, after one iteration, to

0	0	0	0	0
0	0	0	0	0
-1	0	-1	0	0
+1	+1	0	0	0
0	+1	-1	0	0

In this case, DELHEIGHT causes precisely the right correction to remove the spike in the data. Notice, however, that the points around the spike have been perturbed slightly; this oscillatory perturbation is small enough to escape further notice by DSLTHRESH, so remains in the data.

When DSLTHRESH is too small to correct the spike completely on the first pass, the following results are typical. The original data is

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
+24	0	0	0	0

One iteration gave

0	0	0	0	0
0	0	0	0	0
-1	0	-1	0	0
+1	+1	0	0	0
+16	+1	-1	0	0

A second iteration resulted in

0	0	0	0	0
0	0	0	0	0
-2	0	-2	0	0
+2	+2	0	0	0
+8	+2	-2	0	0

The third iteration produced

0	0	0	0	0
-1	0	0	0	0
0	0	-2	0	0
+2	+2	0	0	0
+5	+2	0	-1	0

On the fourth iteration, it converged to

0	0	0	0	0
-1	0	0	0	0
0	0	-2	0	0
+3	+2	0	0	0
+4	+3	0	-1	0

The perturbations in the data are here first reinforced, then serve to substantiate the presence of the error. The spike is reduced to a smoothly oscillating hump, which passes the error thresholds, thus remains in the data.

When DELHEIGHT is too large, so that spikes are over-corrected, the following behaviour can result. Here, DSLTHRESH=.05 and DELHEIGHT=16/12. The original data is

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
+8	0	0	0	0

Successive iterations give

0	0	0	0	0
0	0	0	0	0
-1	0	-1	0	0
+3	+3	0	0	0
-8	+3	-1	0	0

then

0	0	0	0	0
-1	0	0	0	0
+4	0	0	0	0
-1	-1	0	0	0
+8	-1	+4	-1	0

then

-1	0	0	0	0
+2	+1	0	0	0
-7	+1	-1	0	0
+3	+6	+1	+1	0
-8	+3	-7	+2	-1

This example does not converge.

When applied to the real data in the sets ETL and WTB1 (see Appendix A for data set descriptions), these problems were less sharply defined, but were still in effect. The slow spreading seen in the first two examples was evident in the resultant smoothing of contours; often important terrain detail was removed. Spike-like errors, such as the two-point error on the flat area at the lower right of the ETL data set, were smoothed to humps rather than removed. For one parameter setting, the nearly flat area of arroyos in the upper right of the ETL data was thrown into oscillations which grew in extent, very much like the behavior shown in the third example.

It was clear that some modifications would have to be made to the basic algorithm to circumvent these problems. Several additions were tried (See Appendix B for descriptions of some of these.), but none produced the desired degree of improvement.

After considering the matter further, we concluded that there were two separate problems with the correction algorithm suggested by ETL. One was its inability to distinguish between valid and invalid data at adjacent points, thus allowing invalid data to corrupt its neighbors. The second was its inability to generate the correct change in height to rectify terrain errors on the first attempt.

We chose to attack these two problems separately, by first distinguishing valid from invalid data, then performing complete corrections based only on valid data.

SECTION 3

ERROR DETECTION

Before we can reasonably correct the data, we must first establish a measure of reliability for each data point. This permits us to give preference to valid points when forming the corrections.

3.1 Simple Measures of Reliability

We would like to develop a measure of reliability at each point which is a number between 0.0 (signifying that this point can't be trusted) and 1.0 (meaning that this point looks very good). The TL and TD of Equation (2-5) each produce a number between -1.0 and 1.0. The sum of these indicators in Equation (2-6a) is therefore a number between -12.0 and 12.0. Taking the absolute value of this sum, divided by 12.0, gives a number in the proper 0.0 to 1.0 range, but with its sense reversed--1.0 means that all of the tests indicated a bad point, while 0.0 indicates either that there were no objections or that the positive and negative objections cancelled out. Subtracting this quantity from 1.0 produces the desired range and polarity of values.

$$RD[I,J] = 1. - ABS\left(\sum_{K=1}^{4} TL[I,J,K] + \sum_{K=1}^{8} TD[I,J,K] \right)/12. \quad (3\text{-}1)$$

A similar measure can be constructed from the slope constraints if we define

$$TS[I,J,K] = 1.0 \quad \text{if } ABS(SLOPE[I,J,K]) \geq \text{slope threshold} \quad (3\text{-}2)$$
$$= 0.0 \quad \text{otherwise}$$

then form the analog of Equation (3-1)

$$RS[I,J] = 1. - \sum_{K=1}^{8} TS[I,J,K])/8. \quad (3\text{-}3)$$

3.2 Weighted Iteration of Reliabilities

In forming the above reliabilities, we are performing simple averaging of the contributions made by each confidence measure. This is the equivalent of a weighted averaging with all of the weights being equal to 1.0. We know that the confidence measures are not all of the same validity, since the data points which formed them vary in validity. Therefore we should use a true weighted averaging in which the confidence measures each have different weights. For the slope constraints, this would be

$$RS2[I,J] = 1. - \frac{\sum_{K=1}^{8} WS[I,J,K] * TS[I,J,K]}{\sum_{K=1}^{8} WS[I,J,K]} \qquad (3\text{-}4)$$

The weight for a slope measure should be related to the reliabilities of the data points which produced the slope. Equation (3-4) is calculating the reliability of one of these data points, so the information needed is the validity of the other point. Thus we use

$$WS[I,J,K] = RS[I+DI[K],J+DJ[K]] \qquad (3\text{-}5)$$

In reality, Equations (3-3) and (3-4) are both instances of a more general iterative form, reminiscent of the relaxation techniques used by Rosenfeld [Rosenfeld, Hummel, and Zucker, 1976],

$$RSI[I,J,N] = 1. - \frac{\sum_{K=1}^{8} RSI[I+DI[K],J+DJ[K],N-1]*TS[I,J,K]}{\sum_{K=1}^{8} RSI[I+DI[K],J+DJ[K],N-1]} \qquad (3\text{-}6)$$

The RSI[I,J,0] can be initialized to 1.0 (or any other constant) to produce the effect of Equation (3-3). However, if other, a priori knowledge exists about the reliability of the data--such as correlation coefficients from the stereo matching process--these can be used instead as the initialization. (For the ETL data set, the use of their correlation coefficients as initialization did not significantly effect the resulting reliabilities.)

The final slope reliability RS[I,J] is defined to be RSI[I,J,N] after sufficient iterations that significant changes are no longer being made. We have used the criterion that the reliabilities have converged when 99% of them are changing less than .05. On the ETL data set, this criterion is usually met on the third iteration.

Figure 3-1 shows the results of this algorithm on the ETL data set. (See Appendix A for a discussion of the original data and its errors.) Note that this reliability measure finds most of the terrain model to be highly consistent. It successfully points out the error at [18,28], although it spreads it somewhat to the neighbors. It also detects the error at [43,39]. Portions of the "error mountain" at the lower right are detected. This algorithm, however, tends to cast doubt on most of the points in the sharp ridge lines.

A similar iterative reliability measure can be developed for the slope consistency tests. Here we have RDI[I,J,N] =

$$1. - \left| \frac{\sum_{K=1}^{4} WL[I,J,K,N]*TL[I,J,K] + \sum_{K=1}^{8} WD[I,J,K,N]*TD[I,J,K]}{\sum_{K=1}^{4} WL[I,J,K,N] + \sum_{K=1}^{8} WD[I,J,K,N]} \right| \quad (3\text{-}7)$$

The weights for the change-in-slope measures depend on the reliabilities of the three data points which produced the two slopes. Because we are calculating the reliability of one of these points, we need examine only the reliabilities of the other two. We have used

```
WL[I,J,K,N] = MINIMUM( RDI[I+DI[K],J+DJ[K],N-1] ,
                       RDI[I-DI[K],J-DJ[K],N-1] )

WD[I,J,K,N] = MINIMUM( RDI[I+DI[K],J+DJ[K],N-1] ,
                       RDI[I+2*DI[K],J+2*DJ[K],N-1] )
```
(3-8)

These weight terms are produced on the "weak link" theory, i.e. that the reliability of a slope difference measure can be no better than the least reliable elevation which went into it.

As before, the final delta-slope reliability RD[I,J] is defined to be RDI[I,J,N] after it has converged. Figure 3-2 shows the results of this algorithm on the ETL data set. Note that this reliability measure finds almost all of the terrain model to be very consistent. It points out the error at [18,28] without the spreading exhibited by the slope measure. It detects the error at [43,39], and sees nothing greatly wrong with the steep ridges, because they are internally consistent. However, it detects only a few bad points in the error mountain, because these points are wrong in a fashion which produced quite consistent slopes.

For the correction routines, we need to form one reliability at each point from the two we have developed. One way of achieving this is to wait until the RSI and RDI have converged, then combine them. Here we use

```
R[I,J] = SQRT( RSI[I,J,M] * RDI[I,J,N] )
```
(3-9)

where M and N are the iterations at which the RSI and RDI are judged to have converged, respectively. Figure 3-3 shows Figures 3-1 and 3-2 combined in this manner. Note that the downgrading of the steep ridge is lessened, while the edges of the error mountain are still fairly well indicated.

Another alternative is to combine the RSI and RDI at each step in the iterations

```
RI[I,J,N] = SQRT( RSI[I,J,N] * RDI[I,J,N] )
```
(3-10)

and define the overall reliabilities R[I,J] to be RI[I,J,N] after it converges. Interestingly, the reliabilities produced by this algorithm are not significantly different from those of Figure 3-3.

3.3 Parameter Setting

Determination of the reliabilities depends on the settings of the thresholds DSLTHRESH and SLTHRESH. The usual manner of doing this is to select parameters which seem "reasonable", or those which have been shown, by experimentation, to work well--"empirically derived parameters". In an attempt to reduce some of the arbitrariness of parameter selection, we investigated a technique for statistical determination of parameters.

Our method is based on the idea that, for most digital terrain models, there will be a preponderance of good data interspersed with a small percentage of bad. Often, this percentage of bad data can be estimated.

The next step is to histogram the slopes and changes-in-slopes for the entire terrain model. For the ETL data, we get the following table. The % columns report cumulative percentage through that class.

Class Limits	Slope Count	Slope %	D-Slope Count	D-Slope %
0.0-0.1	12510	67.17	5829	64.59
0.1-0.2	3078	83.70	1701	83.44
0.2-0.3	1574	92.15	663	90.79
0.3-0.4	832	96.62	357	94.75
0.4-0.5	364	98.57	181	96.75
0.5-0.6	146	99.36	105	97.92
0.6-0.7	56	99.66	70	98.69
0.7-0.8	22	99.77	42	99.16

If, for example, we feel that the percentage of bad points is approximately 4%, then the suggested parameters for this data set are SLTHRESH=0.4, DSLTHRESH=0.5--the points at which our histogram approximates 96%. These are the parameters we have used for the examples in this chapter.

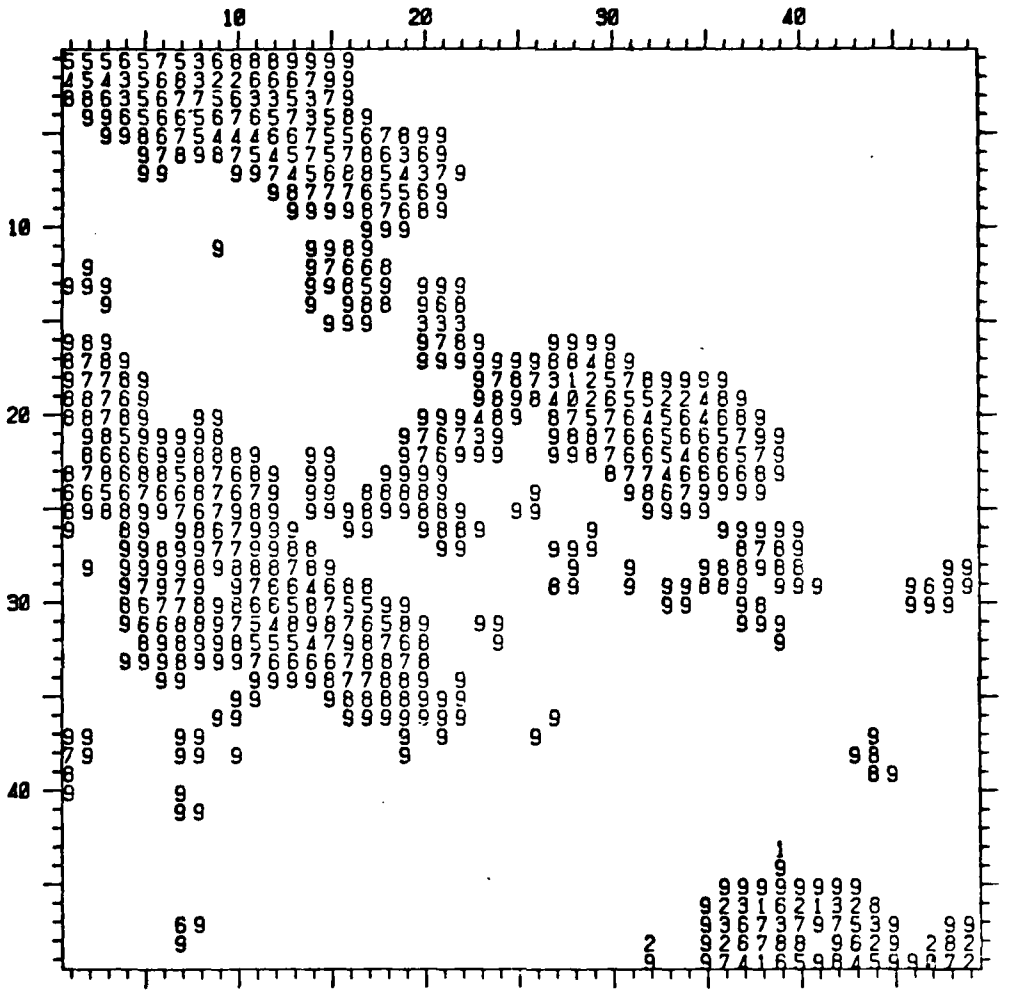


Figure 3-1.

These are the reliabilities for the ETL data set, produced by 3 iterations of slope analysis. Initial reliabilities were 0.5; SLTHRESH=0.4, PCNULL=0.5. Integers are 10 times the reliabilities; blank spaces indicate reliabilities near 1.0.

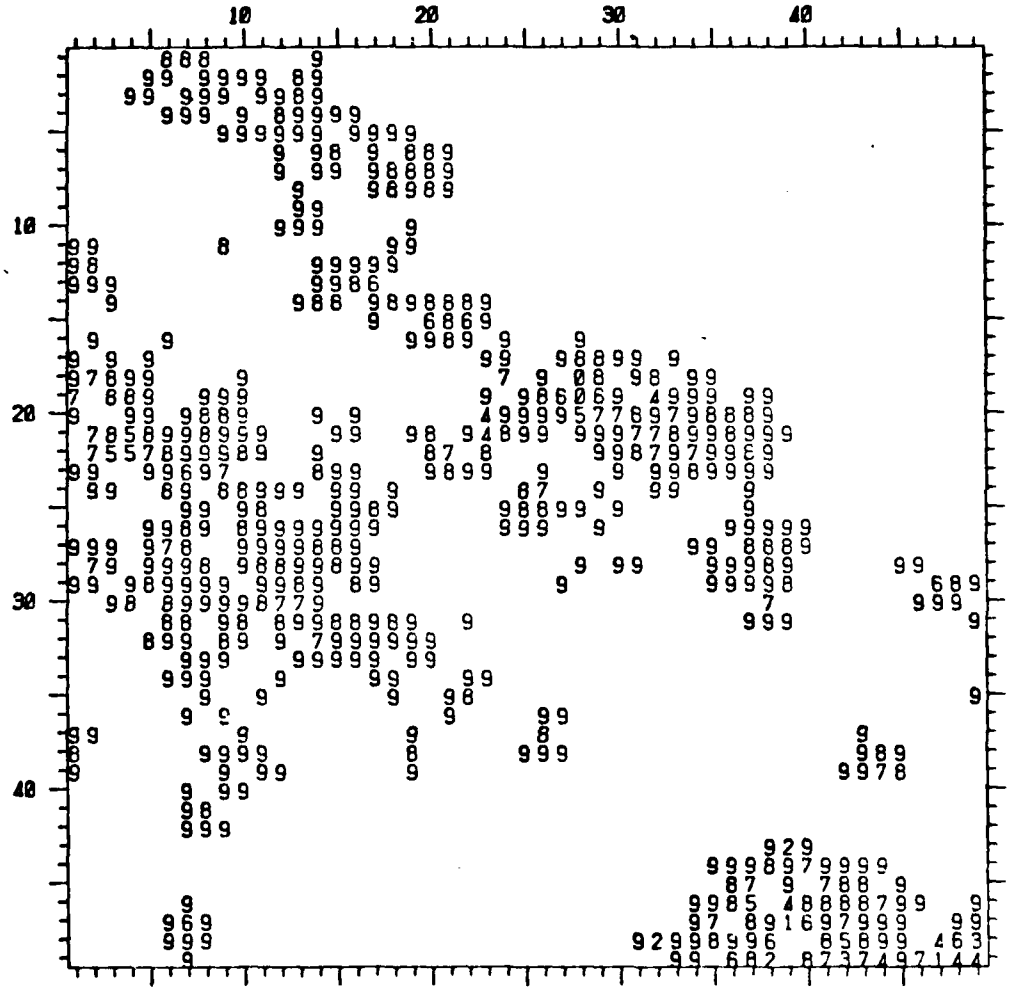


Figure 3-2.

These are the reliabilities for the ETL data set, produced by 3 iterations of change-in-slope analysis. Initial reliabilities were 0.5; DSLTHRESH=0.5, PCNULL=0.5.

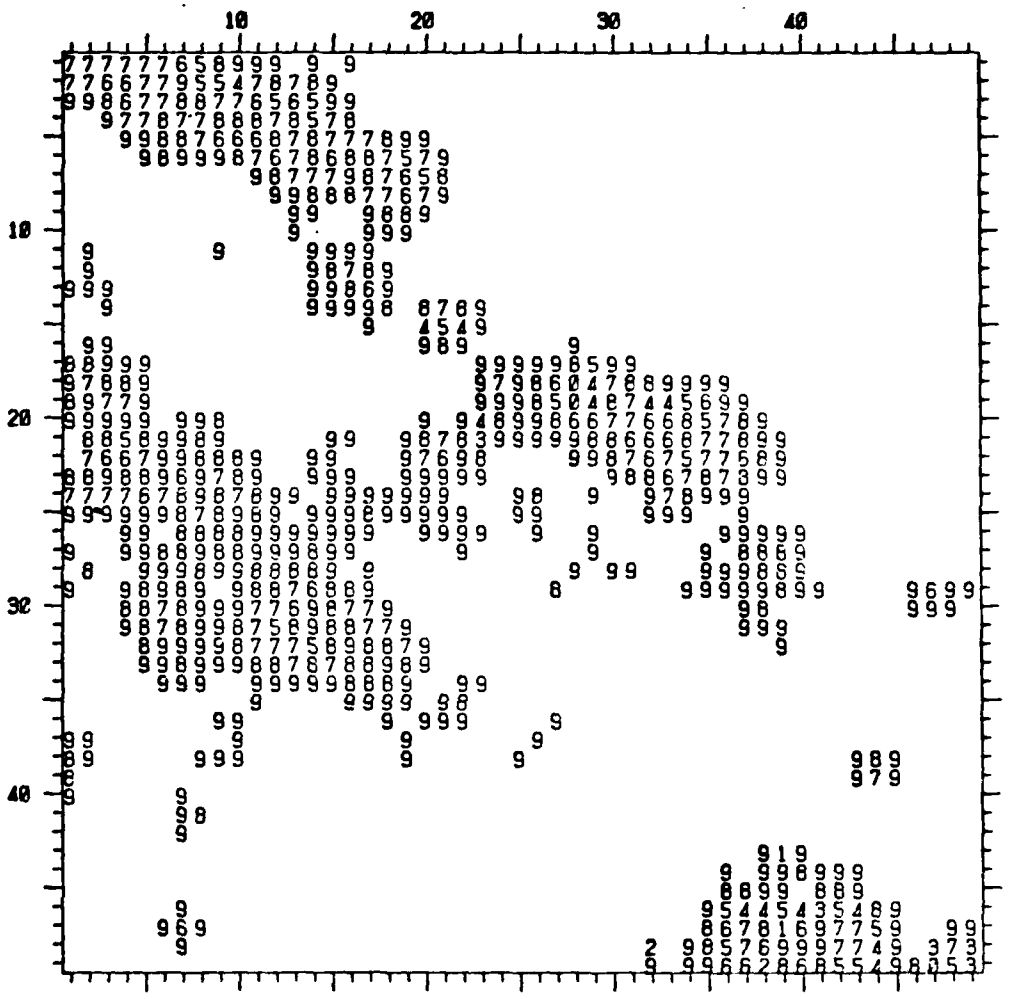


Figure 3-3.

These are the reliabilities for the ETL data set, produced by combining Figures 3-1 and 3-2 in the manner described in Equation (3-9).

SECTION 4

ERROR CORRECTION

In Section 3, we developed reliability measures on the data, suitable for use as weighting coefficients. In this section, we explore methods for using these weights in correcting the data.

4.1 Direct Correction

As we noted in Section 2.2, the desired correction algorithm is one which will produce, in a single step, an elevation which is as compatible as possible with its neighbors, given their reliabilities. This implies that some sort of weighted averaging might be in order.

Simple weighted averaging, however, has an obvious drawback. As we pointed out in Section B.2, elevations at the tops of ridge lines are likely to be lowered if they are replaced by the average of their neighbors; the inverse is true of stream bottoms. Weighting the neighbors does not change this fact, so weighted averaging is not the algorithm we want.

In Section B.2, we also discussed the use of a slope extrapolation technique. We tried combining this technique, as set forth in Equation (B-2), with the weighting technique of Equation (3-8). When we tried this on the PATCH data set, we found that a scan across the edge of the cliff, which originally read

48 48 48 48 48 48 32 32 32 32 32 32

turned into

48 48 48 48 54 42 38 26 32 32 32 32

Further iteration caused the oscillation to spread. Clearly this algorithm is not the one we want, either.

After some thought, we concluded that the iterative algorithm suggested by ETL was attempting to seek at each point the elevation value which maximized the reliability (hence minimized the change-in-slope measure). Consequently, we next attempted replacing each point with the elevation that minimized the weighted average change-in-slope. Mathematically, this involves finding the H'[I,J] which minimizes

$$\frac{\sum_{K=1}^{4} WL[I,J,K]*ABS(DSLOPL[I,J,K]) + \sum_{K=1}^{8} WD[I,J,K]*ABS(DSLOPD[I,J,K])}{\sum_{K=1}^{4} WL[I,J,K] + \sum_{K=1}^{8} WD[I,J,K]} \qquad (4\text{-}1)$$

where the WL, WD, DSLOPL, and DSLOPD are as defined previously, but use the combined R[I,J], with each hypothesized H'[I,J] substituted for H[I,J]. The solution is obtained by predicting the upper and lower bounds on the desired elevation (based on the maximum and minimum elevations extrapolated by the neighboring points), then conducting a binary search to minimize the expression between these limits.

We first tried this algorithm on the PATCH data set. As expected, the noise spikes on the flat and on the ramp disappeared completely on the first iteration. We were surprised to note that the oscillator at the top was also completely removed. (Other algorithms had smeared it to an average value.) Also surprising was the fact that the lower oscillator was not changing internally, but was being "eaten away" at the edges; after 4 iterations it, too, was completely removed. Furthermore, the cliff edge was changing only at its ends--apparently there was enough internal consistency in the planes above and below the edge that the algorithm could only "get a bite on" the ends, where boundary effects lowered the data consistency. With several iterations, a slight smoothing propagated along the edge.

We then tried this algorithm on the ETL data set; Figure 4-1 shows the results. Note that one iteration has removed most of the 1- and 2-point errors, leaving only the very tiny ones. The contours, although somewhat smoothed, do not show the drastic over-smoothing produced by the original ETL algorithm.

The function on which this algorithm is based does not have an analytical solution, but must be minimized by iterative "trial and error" techniques. Consequently, this algorithm is fairly expensive to apply. In an attempt to reduce this expense, we tried a related algorithm which, instead of minimizing the sum of the absolute value of the slope differences, minimizes the sum of the squares of the slope differences--a function which has an analytical solution. Use of this modification got no further than the PATCH data set. It removed the spikes properly, but set the ramp edge into ringing oscillations similar to what the extrapolation methods had produced. Expensive or not, Equation (4-1) produced the desired effect, so this is our algorithm of choice.

4.2 Constraints on Corrections

When every point in the elevation matrix is replaced in this manner, as in Figure 4-1, some smoothing of the contours will result. This can cause the loss of detail, especially in areas of ridge tops and canyon bottoms. To avoid this, constraints have been imposed on the use of this smoothing.

An obvious constraint is to only change an elevation if its reliability is less than some threshold RTHRESH. This, by itself, does not produce satisfactory results.

A second constraint is based on the statistical relationship of the adjusted elevation with its neighbors. To get a measure of the variation in the neighborhood, we first calculate the weighted standard deviation of the elevations of the 8 neighboring points. If we define

$$\text{WMEAN}(H) = \frac{\sum_{K=1}^{8} R[I+DI[K],J+DJ[K]] * H[I+DI[K],J+DJ[K]]}{\sum_{K=1}^{8} R[I+DI[K],J+DJ[K]]}$$

then we can express WSIGMA(H) = (4-2)

$$\text{SQRT}\left(\frac{\sum_{K=1}^{8} R[I+DI[K],J+DJ[K]] *(H[I+DI[K],J+DJ[K]] - \text{WMEAN}(H))^2}{\sum_{K=1}^{8} R[I+DI[K],J+DJ[K]]} \right)$$

We then compare the suggested change in elevation (the original elevation minus the newly calculated one) against this WSIGMA,

$$\text{ABS}(H[I,J] - H'[I,J]) > \text{SIGCONST} * \text{WSIGMA}(H) \qquad (4\text{-}3)$$

(SIGCONST allows us to turn this feature up or down; it is usually 1.0.) We then implement the change only if it is larger than this local variability measure. The rationale behind this constraint is that if the change is large, it is likely that the original data is in error, so the change should be implemented. However, if the change is small, then the original data is hypothesized to represent terrain detail rather than error, and the original data should be retained.

Figure 4-2 shows the results of our correction algorithm when constrained by both of these suggestions, that is, implementing a change if the reliability is less than RTHRESH or if the change is greater than SIGCONST*WSIGMA, for RTHRESH=0.5 and SIGCONST=1.0. Here, all of the major 1- and 2-point errors have been detected and reasonably corrected. A few small contours remain, and a few of the contours are a bit rough, but the original terrain detail is intact. The error mountain is still there, but it has been compressed into a steep anomaly that the reliability measure can continue to detect.

Having changed the elevation data with this correction algorithm, it is possible that some of the points adjacent to corrected points will now need adjustment. The total algorithm--iterating to find the reliability measures, then correcting the data--can itself be iterated, until sufficient correction is deemed to have been made.

These algorithms are sufficiently complicated that no proof of convergence is practical. Since we are selecting elevations which will minimize the change-in-slope, it is reasonable to expect that the algorithm will eventually converge. On the PATCH data set, the unconstrained algorithm had ceased to make meaningful changes after four iterations. On the ETL data set, iterations beyond the first pass of the unconstrained algorithm produced mostly small, unimportant changes. The constraints, which limit the number of points changed, expedite this convergence. No occurrances of oscillatory behaviour have been observed, although, again, it cannot be proven that oscillation is impossible.

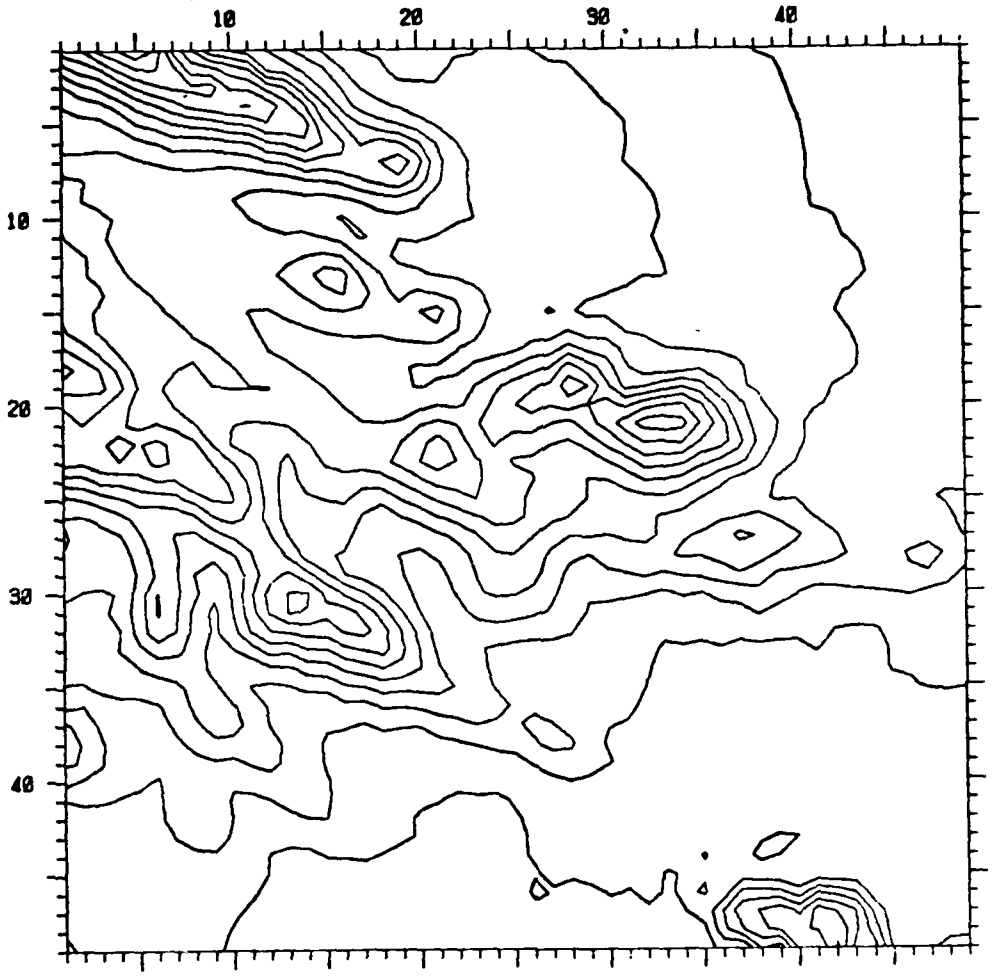


Figure 4-1.

This figure shows the ETL data set after 1 iteration of the change-in-slope minimization algorithm, without any constraints. Parameters for the run were: confidences initialized to 0.5, DSLTHRESH=0.5, SLTHRESH=0.4, PCNULL=0.5, RTHRESH=0.0, SIGCONST=0.0. Elevations range from 363 meters at [49,16] to 527 meters at [1,5]; contours are at 10 meter intervals.

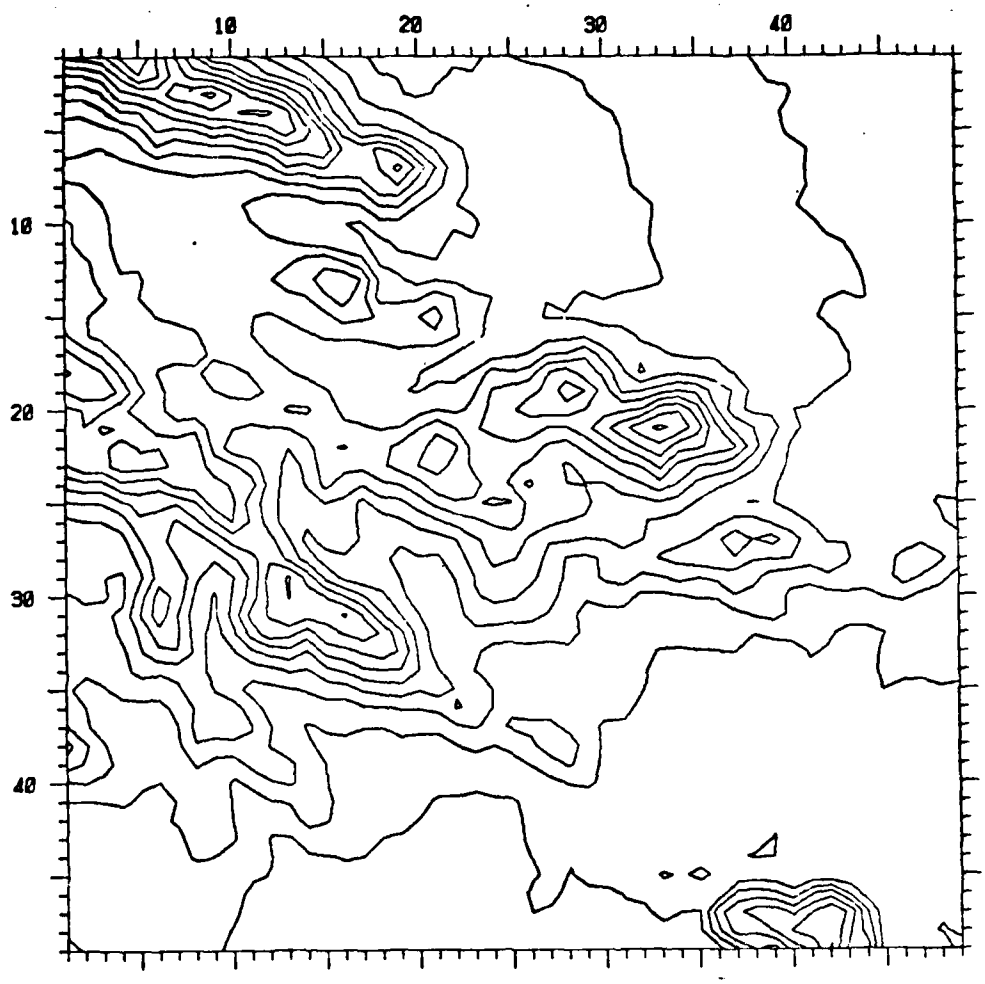


Figure 4-2.

This figure shows the ETL data set after 1 iteration of the change-in-slope minimization algorithm, with correction constraints. Parameters for the run were: confidences initialized to 0.5, DSLTHRESH=0.5, SLTHRESH=0.4, PCNULL=0.5, RTHRESH=0.5, SIGCONST=1.0.

SECTION 5

LAND-TYPE CLASSIFICATION INFORMATION AS AN AID

Under this contract to ETL, we also made an investigation into the use of land-type classification data to influence the elevation error detection and correction processes. For this analysis, ETL provided an array of land-type classifications for each elevation point in the terrain model. This classification data is shown in Figure 5-1.

The simplest way to allow classification data to influence the detection and correction processes is to permit it to influence the parameters which control the processes. The most sensitive of these are the evaluation thresholds, DSLTHRESH and SLTHRESH. Consequently we implemented these two parameters as table look-up functions of the classification for each point.

$$\text{DSLTHRESH} = \text{DSLTHR}[\ \text{CLASS}[I,J]\ ] \qquad (5\text{-}1)$$

$$\text{SLTHRESH} = \text{SLTHR}[\ \text{CLASS}[I,J]\ ]$$

These parameters are set in the statistical fashion described in Section 3.3. Instead of forming the single histogram shown there, we form N separate ones, one for each classification. Thus for each point, we first look up its classification to determine which histogram to use, then calculate the slopes and changes in slopes, and attribute them to the appropriate bins of that histogram. Parameters for each class are then chosen independently, from that class's histogram.

Attempts to use the ETL classifications to influence the data met with mixed success, mainly because these classifications pertain more to ground cover than to elevation, slope, and roughness information. Some of these classes can be used to infer surface data; for instance, 2-lane roads, orchards, drainage canals, and buildings could be expected to lie on fairly flat ground. The class for dry creeks, however, is used for both the flat area of arroyos at the upper right of the model and for the steep canyons in the hilly area. Likewise, the class for dirt roads does not distinguish between agricultural roads in the orchard and jeep trails in the hills. To compound the problem, the classifications are not without error. In rows 33, 34, and 35 between columns 12 and 19, there is a dry creek marked which follows a contour along a rather steep hillside. Examination of the aerial photo lead us to believe this is an outcropping of rimrock, with bushes.

We felt that the idea of using outside knowledge as to the land-type deserved a fair trial. Consequently, we set about deriving from the ETL classifications and the elevation data set itself a new set of land-type classifications. For each point, we first examined the ETL classification. If it indicated a 2-lane road, an orchard, a drainage ditch, or a building, we reclassified the point as being flat. If the point was a dirt road, then we examined the neighboring classes; if the

road adjoined one of our flat classes, then that point was reclassified as being flat. Points which did not meet either of these criteria were classified by the use of two roughness measures, the standard deviation, and the deviation from a least squares plane, shown in Equation (B-5). Rather arbitrarily, we reclassified as flat those points with ordinary-SIGMA less than 2.; points with planar-SIGMA greater than 5. were reclassified as rough; everything else fell into the medium slope category. This produced fairly reasonable results, with a few exceptions around the spike errors in the lower right area. In the interests of a fair test of the knowledge base, these points were corrected by hand. Figure 5-2 shows the classifications which resulted.

We then tried the algorithm of Sections 3.2 and 4.2 on the ETL data set, using appropriate parameters for these classifications. Figure 5-3 shows the reliability measures which resulted. Most of the model is found to be quite consistent, although there is some degradation in the steep hills. The errors, especially the spikes in the flat and the error mountain, are clearly marked. The results of one iteration of correction are shown in Figure 5-4. These contours do not differ greatly from the results of Figure 4-2; however, after eight iterations of the algorithm, we obtained Figure 5-5. Notice that the error mountain in the lower right corner has been completely removed. Without this additional classification information, we were not able to duplicate this feat.

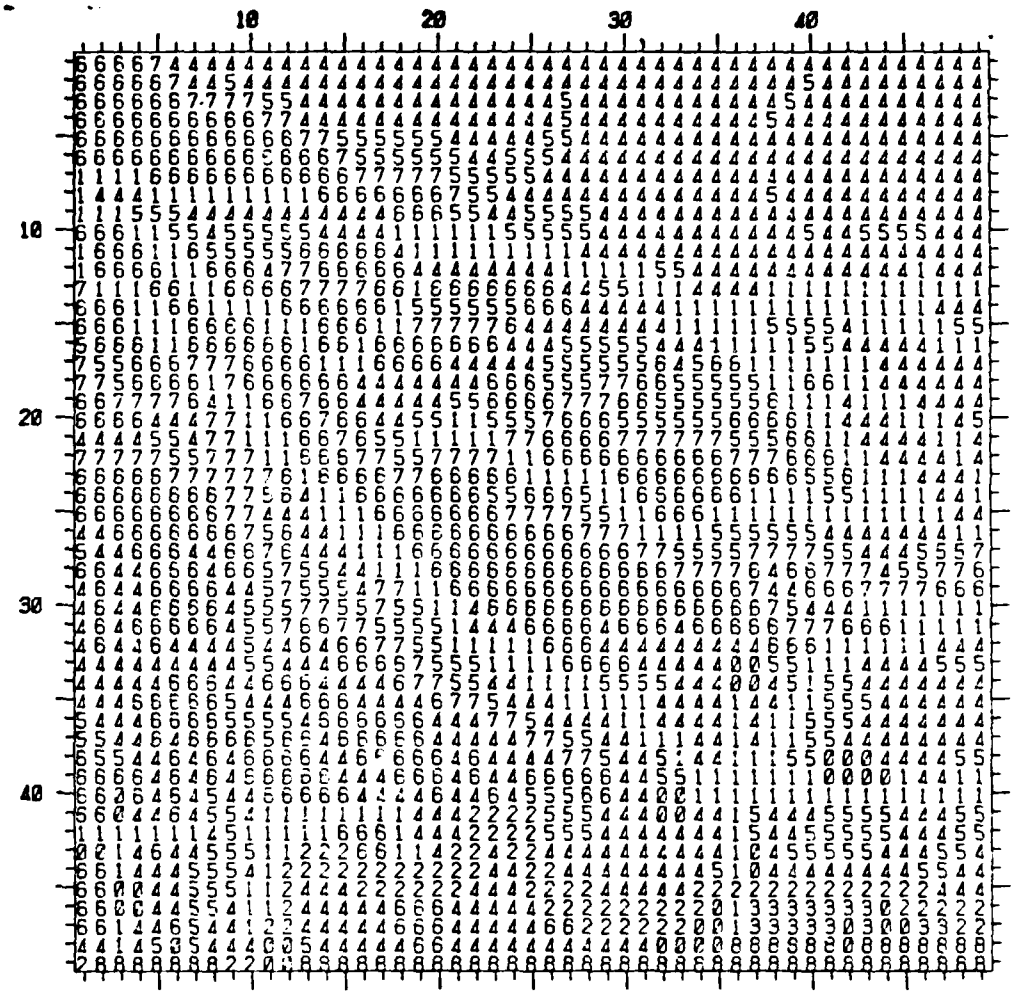


Figure 5-1.

This figure shows the land-type classification data provided by ETL. Classes are:

0 Buildings
1 Trails or Dirt Roads
2 Drainage Canal
3 Orchard
4 Dry Creek with Bushes
5 Bare Hillside
6 Hillside with Bushes
7 Ridge
8 2-Lane Primary Road

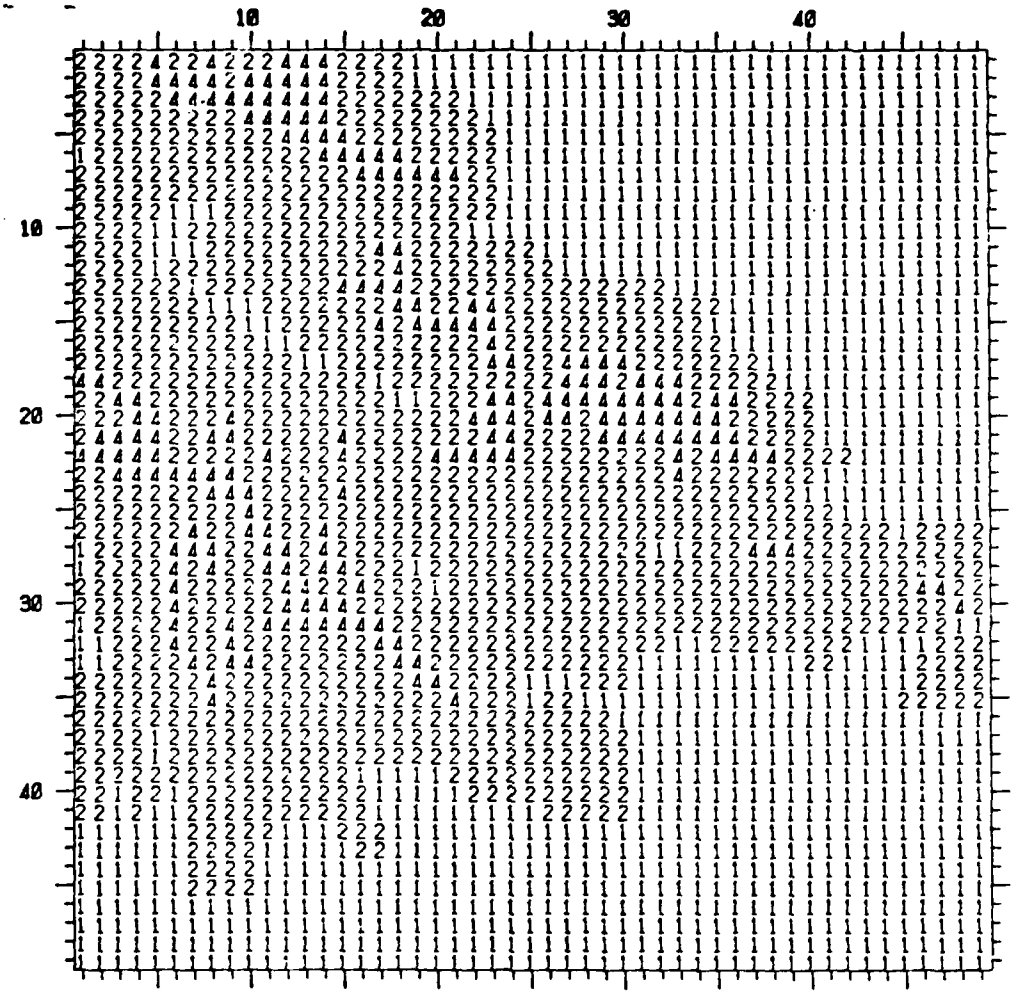


Figure 5-2.

This figure shows the land-type classification data which we derived. Classes are:

1 Flat Areas
2 Medium Sloped Areas
4 Very Rough Areas

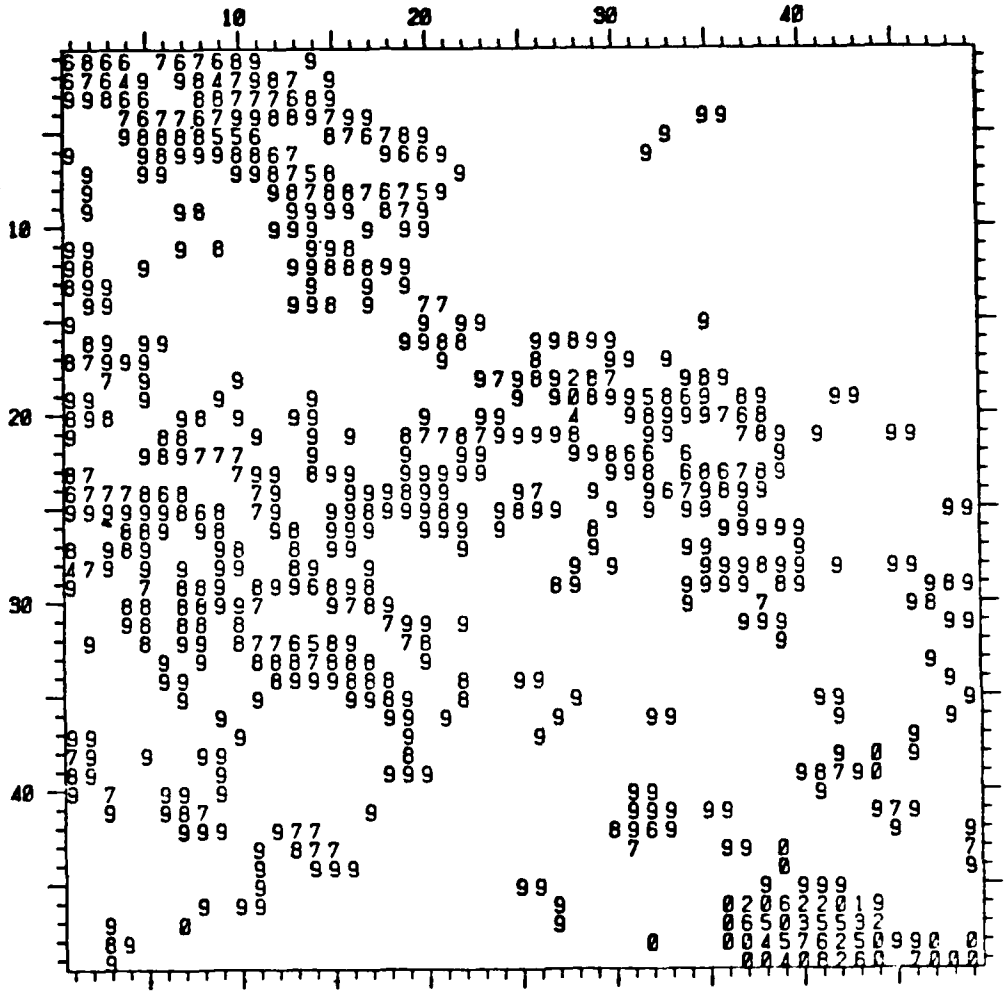


Figure 5-3.

This figure shows the reliabilities for the ETL data set, resulting from 3 iterations of the combined slope and change-in-slope analysis, using the derived classification data. Initial reliabilities were 0.5, DSLTHRESH=[0.2,0.3,0.8], SLTHRESH=[0.1,0.3,0.5], PCNULL=1.0.

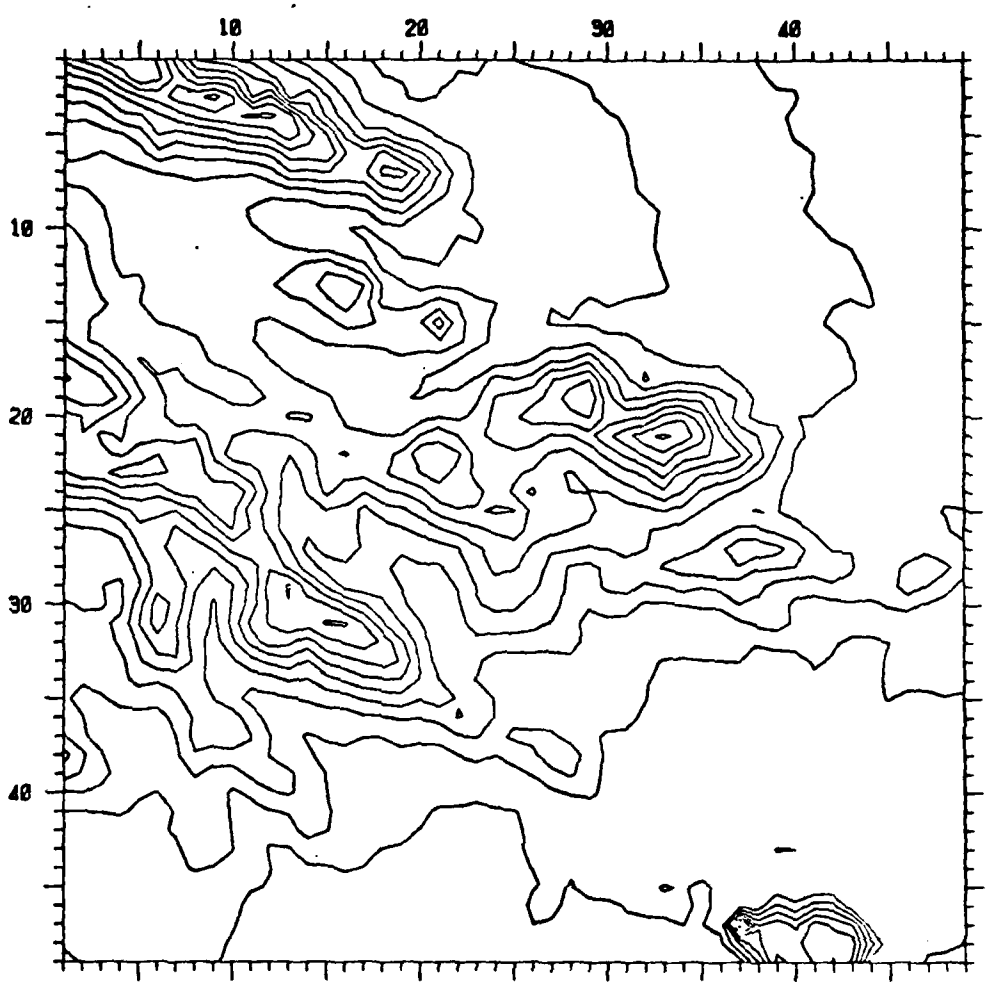


Figure 5-4.

This figure shows the ETL data set after one iteration of the change-in-slope minimization algorithm, with correction constraints and using the derived classification data. Initial reliabilities were 0.5, DSLTHRESH=[0.2,0.3,0.8], SLTHRESH=[0.1,0.3,0.5], PCNULL=1.0, RTHRESH=0.5, SIGCONST=1.0.

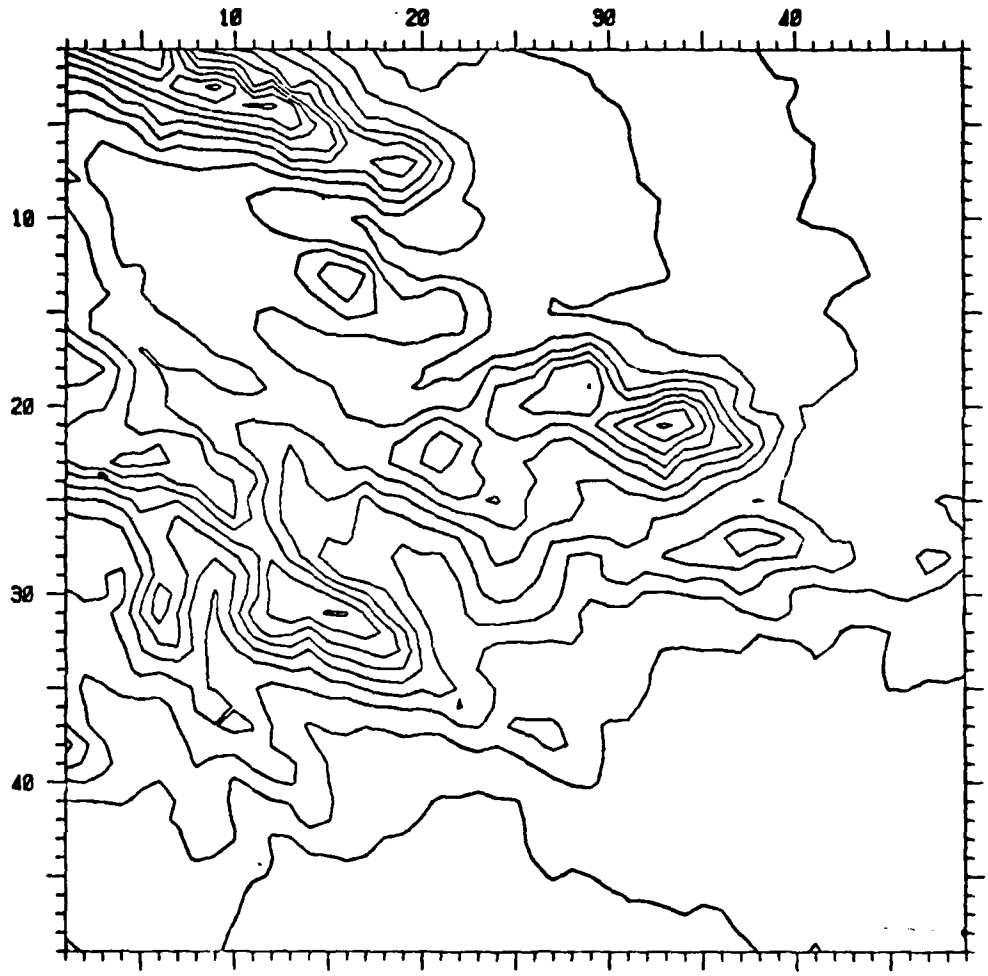


Figure 5-5.

This figure shows the ETL data set after eight iterations of the change-in-slope minimization algorithm, with correction constraints and using the derived classification data. Initial reliabilities were 0.5, DSLTHRESH=[0.2,0.3,0.8], SLTHRESH=[0.1,0.3,0.5], PCNULL=1.0, RTHRESH=0.5, SIGCONST=1.0.

SECTION 6

CONCLUSIONS

Under this contract to ETL, we have developed a method for detecting errors in digital terrain models and an algorithm for correcting these errors. We have also investigated the use of land-type classifications in influencing these processes. In this section we present our conclusions from the results we have obtained.

We began with an algorithm suggested by ETL which used the consistency of each point with its neighbors in determining how to correct that point. Work with this algorithm convinced us that the data correction task was really two subtasks--1) detection of points which were in error and 2) correction of the errors. To this end, we separated the tasks, so that we first distinguished valid from invalid data, then performed corrections based on the valid data points.

The error detection algorithm combines thresholding of slope and of change-in-slope information in an iterative weighted fashion to produce a measure of the reliability of each elevation point. This algorithm handles all points identically, so is highly suitable for implementation on a parallel processor such as IAC's ILLIAC 4 or ETL's STARAN. The algorithm is fairly efficient; a breadboard implementation on IAC's TENEX system requires approximately 20 seconds of CPU time for the 3 iterations required for the reliabilities to converge on the ETL data set. The indications given by these reliabilities are somewhat parameter dependent, but a statistical method for chosing adequate parameters has been developed and presented.

The error correction algorithm selects replacement elevations by finding the minimum of a change-in-slope measure at each point. This algorithm is fairly well suited for parallel implementation; the search for a minimum in the inner loop should be reformulated if a parallel implementation is desired. The algorithm is, unfortunately, not very efficient--our breadboard implementation requires over 90 seconds of TENEX CPU time for a single iteration of correction on the ETL data set.

The correction algorithm is both robust and flexible. It will produce reasonable terrain corrections over a wide range of parameters to the error detection process, freeing the user from excessive worry about precise parameter selection. The bare algorithm, unconstrained in any way, tends to smooth the contours of a terrain model. For some models, this is a desirable effect. For models with more widely spaced points, such as the ETL data set, this can result in the removal of terrain detail. The constraints which have been developed for this algorithm permit the user to control the amount of smoothing desired.

The recommended use of these techniques is as follows. For most digital terrain models, a single iteration of the error detection and correction process should be performed. This will remove small errors--those

consisting of only 1 or 2 points and those of narrow extent. A second pass of the error detection scheme should be then applied, to assess the resulting model. Large areas of points with low reliabilities should be dealt with separately, perhaps by having the correlation algorithm go over these areas again, given their now corrected context; or perhaps they should be hand corrected. It is not recommended that the correction algorithm be iterated more than 2 or 3 times, as the ratio of payoff to cost diminishes severely after the first iteration. Also, correction of large errors is risky by this method, since the algorithm is, in essence, extrapolating the surrounding terrain into the void caused by the error. The results of such large-scale extrapolation are known to be unreliable.

When exterior knowledge in the form of land-type classification is available, it can significantly improve the error detection process. This improvement allows the error correction process to handle even large errors in the data in a fairly reasonable fashion. The land-type data, however, must be directly related to the land-form types, and must be correct. Techniques should be developed to correct machine-derived classifications, both separately and in conjunction with elevation data.

APPENDIX A

DATA SETS

The algorithms described in this research were developed using three sets of data. This appendix describes and illustrates these data sets. (In what follows, co-ordinates on the maps, such as [I,J], refer to the point in row I and column J.)

A.1 The PATCH Data Set

The first data set on which any algorithm was tried was an artificial one, called PATCH, shown in Figure A-1. The data set is divided into 4 areas or patches (hence its name).

The base of the data set is a flat plane at level 32, except in the upper right quarter, where the base is a plane sloping from 32 on the left to 62 at the right edge. The interface between the sloped plane at the upper right and the flat plane at the lower right is a cliff which gradually increases in height from left to right.

The upper left quarter has an oscillator in its center, consisting of an area of the form

```
32 32 32 32 32 32 32
32 40 32 40 32 40 32
32 32 32 32 32 32 32
32 40 32 40 32 40 32
32 32 32 32 32 32 32
32 40 32 40 32 40 32
32 32 32 32 32 32 32
```

The lower left quarter contains a similar oscillator of the form

```
32 32 32 32 32 32 32 32
32 40 32 40 32 40 32 32
32 32 40 32 40 32 40 32
32 40 32 40 32 40 32 32
32 32 40 32 40 32 40 32
32 40 32 40 32 40 32 32
32 32 40 32 40 32 40 32
32 32 32 32 32 32 32 32
```

Both the upper and lower right quarters have had several noise spikes superimposed on them. These features, along with the cliff and the oscillators, provide ample opportunity for an algorithm to show us its less desirable traits. Indeed, several algorithms were discarded after making a poor showing on this data set.

A.2 The USGS Data Set--WTB1

When we first began our research, ETL had not yet sent us a data set. The IAC archives contained a digital terrain model of the White Tail Butte, Wyoming, 7.5' quadrangle, which had been sent to us under an earlier digital terrain model processing contract with USGS. For this report, we use a window of data out of the southeast corner of the quadrangle. It is designated WTB1, and is shown in Figure A-2.

The terrain of White Tail Butte quad is fairly rough, consisting of a number of buttes cut by rugged canyons. The buttes are separated by broad, flat valleys, in contrast to the steep slopes surrounding them. The chosen area is crossed diagonally by one of these flat valleys, with steep buttes on either side.

USGS characterized this data set as noisy. There are many small closed contours cluttering up the flat areas, and many of the contours have a rough, jagged appearance. The data set is fairly typical of stereo derived elevation data over an area with low grey-level contrast.

A.3 The ETL Data Set

The digital terrain model we received from ETL was of an area near Phoenix, Arizona, and is shown in Figure A-3. This model also provides a selection of terrain types.

The major feature of the area is the end of a very rough range of hills, occupying the upper left 2/3 of the model. To the right of this is an area of fairly flat arroyos; below it is an agricultural area, bounded by an irrigation canal and a highway. The lower right corner contains an orchard, visible on both USGS topo maps and on the aerial photos provided us by ETL.

The hilly part of the model is fairly consistent. There are a few places, such as in the ridge at the very top, where the contours contain "glitches", due to mismatched points. There are a few short closed contours, as well. The most glaring error here is a large depression in the middle of the peak which is just above and to the right of center (co-ordinates [18,28]). The area of arroyos is fairly clean, having only a few short or jagged contours.

The agricultural area at the bottom of the model is rather messy. There are a fair number of small contours, representing noise spikes, and the contour which crosses the area has a number of strange crooks in it. The area occupied by an orchard on the aerial photo is covered in the terrain model by a fairly large, steep hill--improbable in the light of the highly regular orchard, the irrigation canal, and the highway which are all in that vicinity. (We have nicknamed this feature Error Mountain.) This, clearly, is a model in need of correction.

This data set was provided with (X,Y,Z) triples for each data point. These were formed by laying down a grid of points in one image, finding

their matches in the second image, then determining the 3-D points which correspond to these matches. The (X,Y) portion of the data thus approximates a grid, but is perturbed slightly at each point by the relief. To simplify our calculations, we have assumed that the data is on a true grid, and have ignored the (X,Y) data.

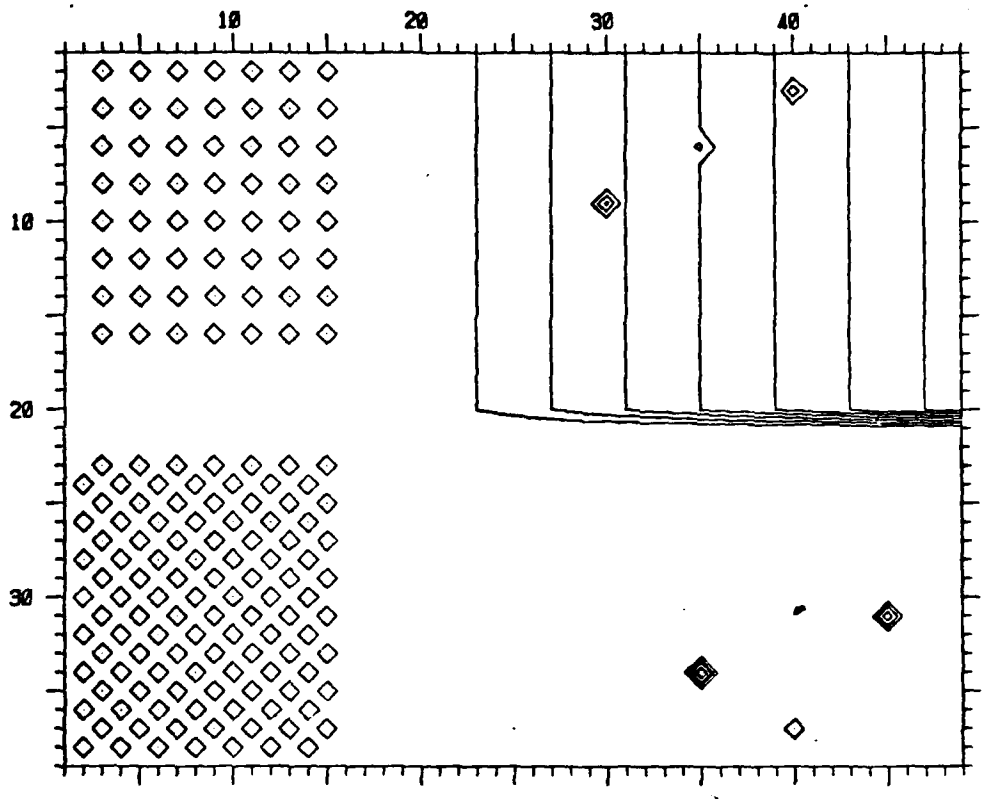


Figure A-1.

This is a contour map of the PATCH artificial data set. The data points are on a nominal 100' grid spacing; elevations range from 32' at [1,1] to 62' at [1,49]; data resolution is 1'. Contours are at 4' intervals.

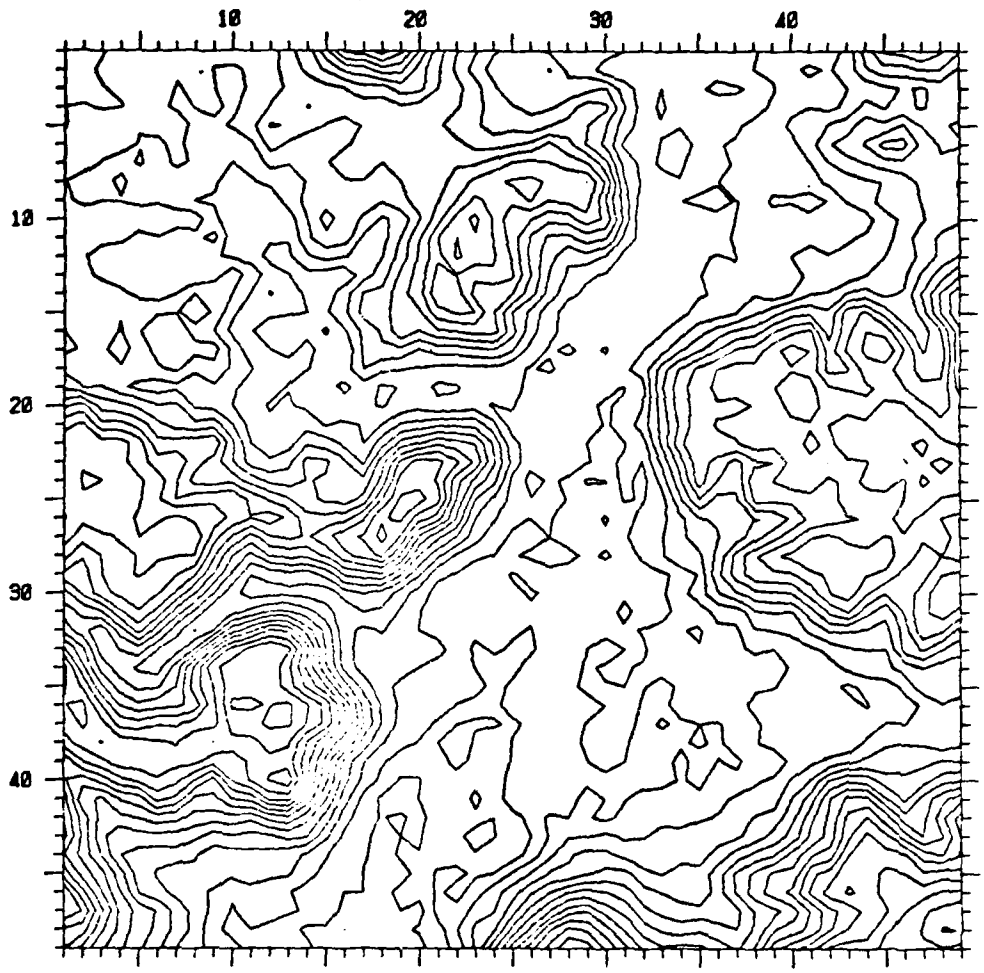


Figure A-2.

This is a contour map of the WTB1 data set. The data points are on a 50 meter (164') grid. Elevations range from 3843' at [49,16] to 4202' at [41,12]; data resolution is 1'. Contours are at 20' intervals.

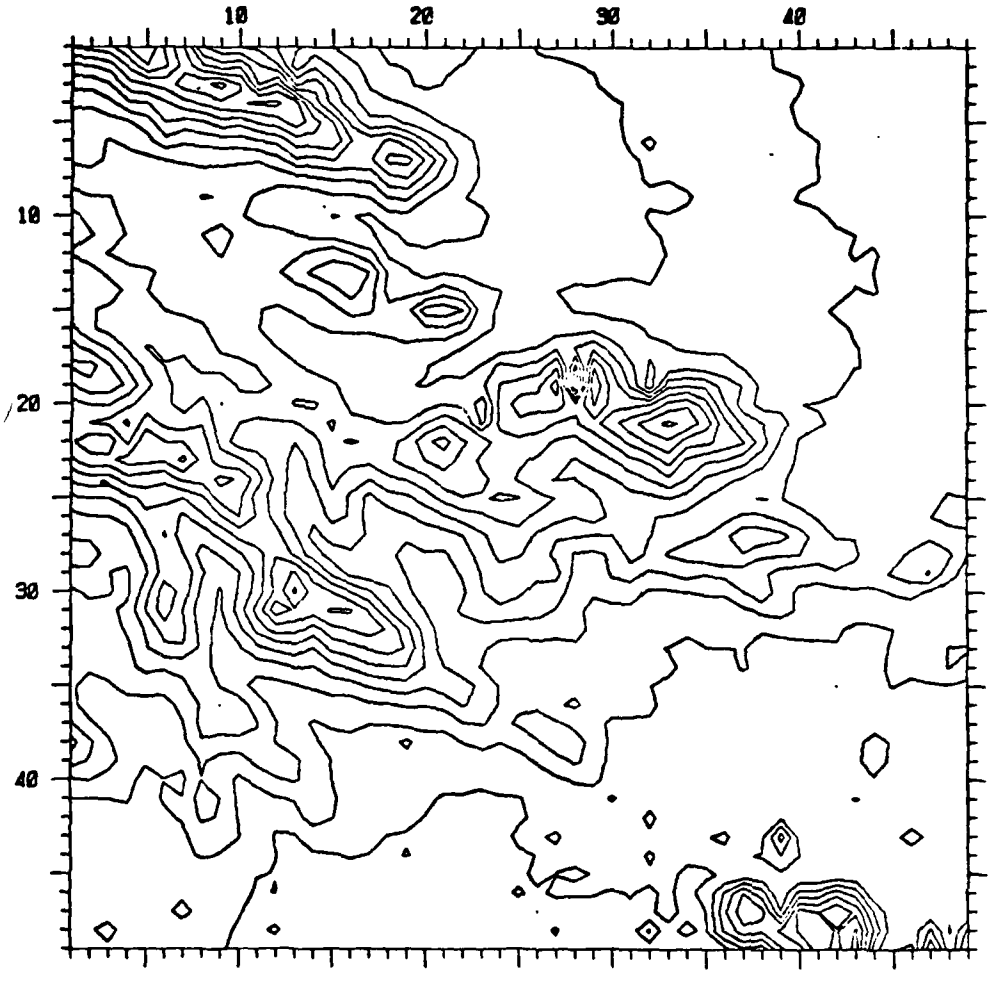


Figure A-3.

This is a contour map of the ETL data set. The data points are on an approximate 45 meter grid. Elevations range from 345 meters at [43,39] to 530 meters at [1,5]; data resolution is 0.1 meters. Contours are at 10 meter intervals.

APPENDIX B

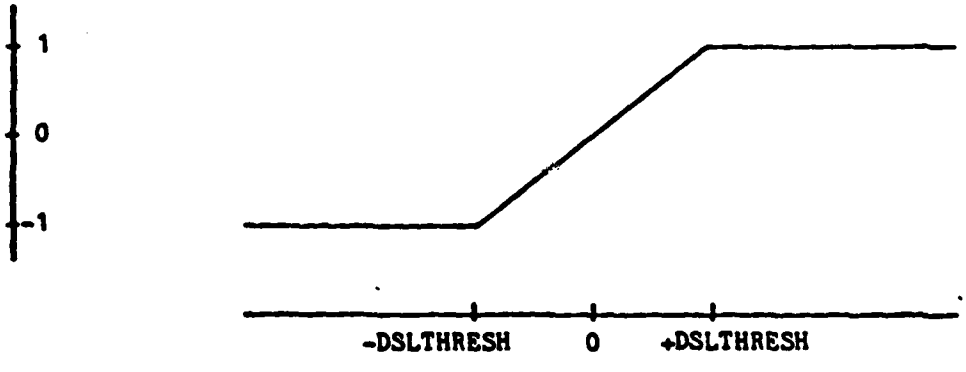
VARIATIONS ON THE BASIC ALGORITHMS

In attempts to remove undesirable effects from the basic algorithm, we tried several variations on the themes suggested to us by ETL. This appendix documents those innovations.

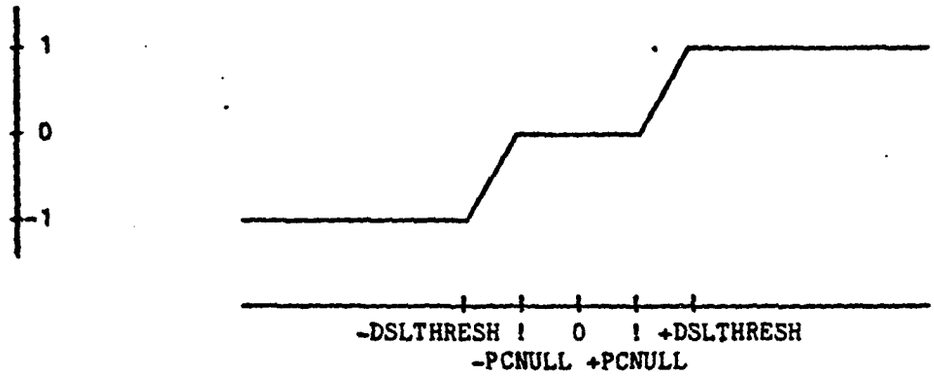
B.1 Threshold Modifications

In ETL's proposed algorithm, each slope comparison generates a vote on the adjustment to the value at the point in question. This vote equals +1 if the difference in slopes exceeds DSLTHRESH (ie, the point should be adjusted upward), 0 if the absolute difference does not exceed DSLTHRESH, or -1 if the difference exceeds -DSLTHRESH (ie, the point should be adjusted downward). This graphs as

This function detects and adjusts points which are highly inconsistent with at least one of their neighbors, but does nothing to points which are slightly inconsistent with most of their neighbors. In the case of the White Tail Butte data set, many of the noise points in the data were missed by this thresholding technique, so we also tried



which permits a point to be changed if several of its neighbors complain about it a little bit. These 2 functions are extreme cases of



The ETL algorithm with this modification was run on the WTB1 data set; Figure B-1 shows the results. The central canyon bottom shows considerable improvement, with much of the contour noise reduced. The canyon walls show a large degree of smoothing, however, and a fair amount of the small side-channel detail is lost.

B.2 Extreme Error Pre-Correction

We next addressed the problem of large spike errors, which are spread into consistent, but incorrect, humps. It was felt that such errors should be detected and removed in advance of the processing.

Detection of the spike errors was accomplished by a technique known in image processing as "Custering" [Quam, 1971]. This is a statistical method which postulates that the data is moderately consistent over local areas. It calculates the expected value of the data (the mean) and the expected variation (the standard deviation, sigma) over a local area around a point, then compares

$$\text{ABS}(H - \text{MEAN}(H)) > \text{CONST} * \text{SIGMA}(H) \qquad \text{(B-1)}$$

where CONST is a constant signifying the cutoff level at which a point is to be considered bad. In our detections, we utilized a 3x3 area surrounding each point and a cutoff of 2.0.

Once a point has been detected, it must be plausibly corrected. An early version of the algorithm simply replaced each point by the mean of its neighbors. This, however, tended to squash the peaks of hills, which were often marked as being at odds with their neighbors. More reasonable results were obtained by averaging the extrapolations of the surrounding slopes coming into a point,

$$H'[I,J] = \sum_{K=1}^{8} (2*H[I+DI[K],J+DJ[K]] - H[I+2*DI[K],J+2*DJ[K]]) \qquad \text{(B-2)}$$

Figure B-2 shows the results of two iterations of this latter algorithm. Note that the huge depression at [18,28] has been plausibly corrected,

that many of the small contours in the lower part of the data set have been removed, and that the hole at [43,39] has been mostly filled. Other small errors and the major systematic error in the lower right corner have escaped the detection process.

B.3 Change-in-Height Modifiers

The obvious drawback of the single DELHEIGHT constant in Equation (2-6a) was that it could not respond to the variations in terrain roughness exhibited by different areas. What was needed was a measure of the variability of the elevations in an area, to be used in tempering DELHEIGHT.

Once again, we used the standard deviation of the elevations in an area as a measure of variability or roughness. At each point, we calculated

$$\text{SIGMA(H)} = \text{SQRT}(\text{MEAN(H}\hat{}\text{2)} - \text{MEAN(H)}\hat{}\text{2}) \qquad \text{(B-3)}$$

Then modified Equation (2-6b) to read

$$\text{H'[I,J]} = \text{H[I,J]} + \text{SIGMA(H[I,J])}*\text{DH[I,J]} \qquad \text{(B-4)}$$

This had the desirable effect of making the corrections respond to the terrain roughness--nearly flat areas received small corrections, while rougher areas got larger corrections. The algorithm still had a tendency to spread peaks rather than correction them, and to perform too much smoothing on detail in the terrain.

The problem with oscillations from too large corrections continued to be a possibility, should DELHEIGHT be set too high. Even for parameter settings which were otherwise reasonable, some portions of the data set sometimes went into mild oscillations along contours on steep hillsides.

After some thought, we determined that this problem stemmed from our variation measurement. The standard deviation, in essence, fits a least squares horizontal plane to the data, then measures the deviations from this plane. In a hillside area, a horizontal plane is a poor fit to the data, and the variability measure has little to do with how a particular point differs from its neighbors. To correct this situation, we implemented a variability measure which first fits a least squares generalized plane to the data, then measures the vertical deviation of the data from this plane.

For this measure, we replace Equation (B-3) with (B-5)

$$\text{SIGMA(H)} = \text{SQRT}\left(\text{MEAN(H}\hat{}\text{2)} - \text{MEAN(H)}\hat{}\text{2} - \frac{\text{MEAN(DI*H)}\hat{}\text{2}}{\text{MEAN(DI}\hat{}\text{2)}} - \frac{\text{MEAN(DJ*H)}\hat{}\text{2}}{\text{MEAN(DJ}\hat{}\text{2)}} \right)$$

and proceed as before. Figure B-3 shows the results of 10 iterations of this algorithm on the data set of Figure B-2. The extreme smoothing and loss of detail indicates that this algorithm is still not the one we want.

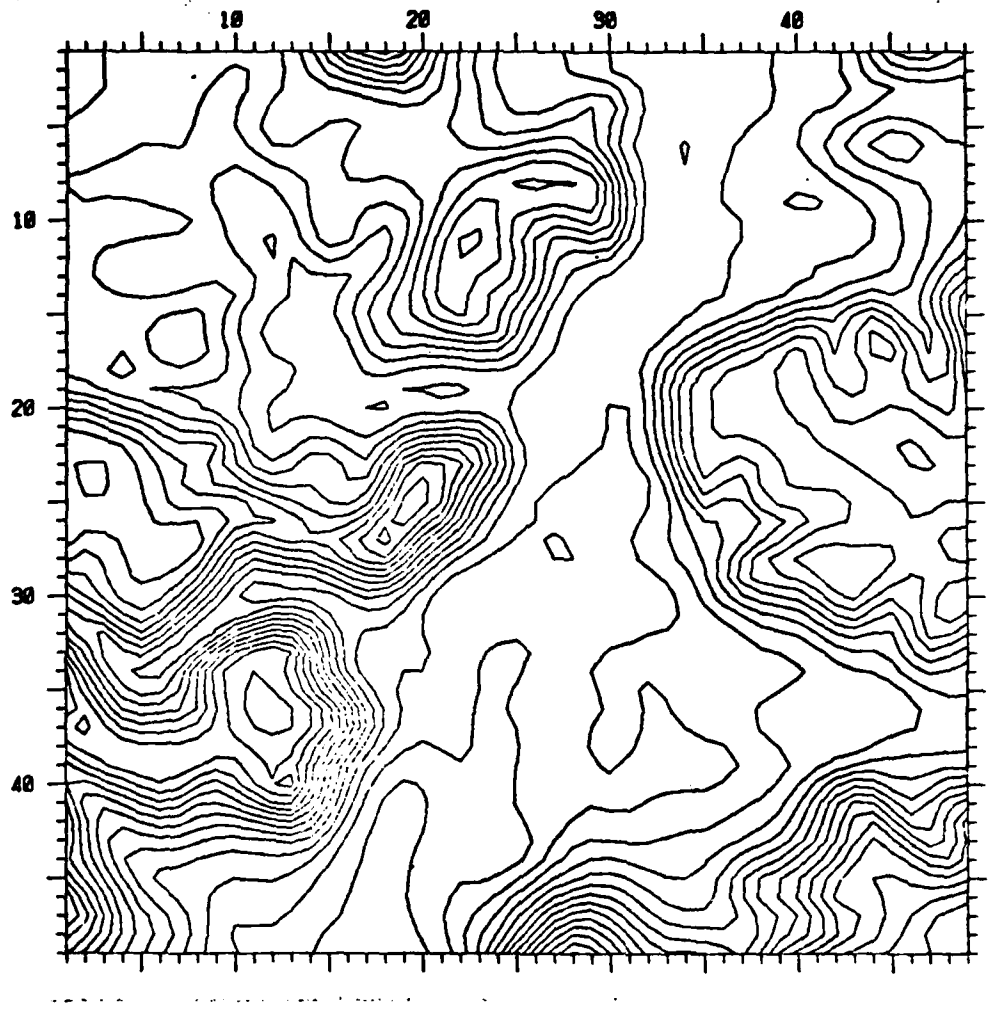


Figure B-1.

This figure shows the portion of the White Tail Butte data set known as WTB1 after 2 iterations of the algorithm described in Equations (2-5) and (2-6), as modified in Section B-1. Parameters for the run were: DSLTHRESH=20/164, DELHEIGHT=12/12, PCNULL=0.0. Contours are at 20' intervals.

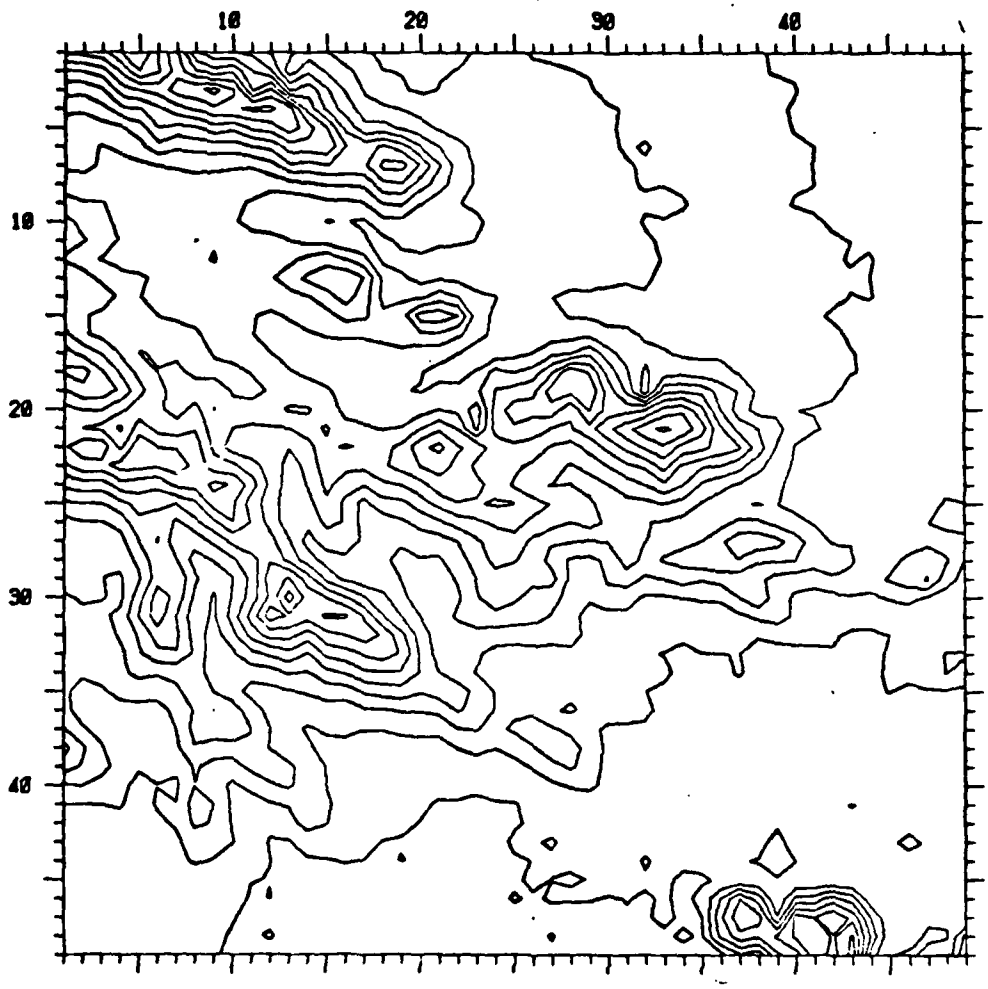


Figure B-2.

This figure shows the ETL Phoenix data set after 2 iterations of spike detection and correction as described in Equations (B-1) and (B-2). Parameters were: area = 3x3, cutoff constant = 2.0. Contours are at 10 meter intervals.

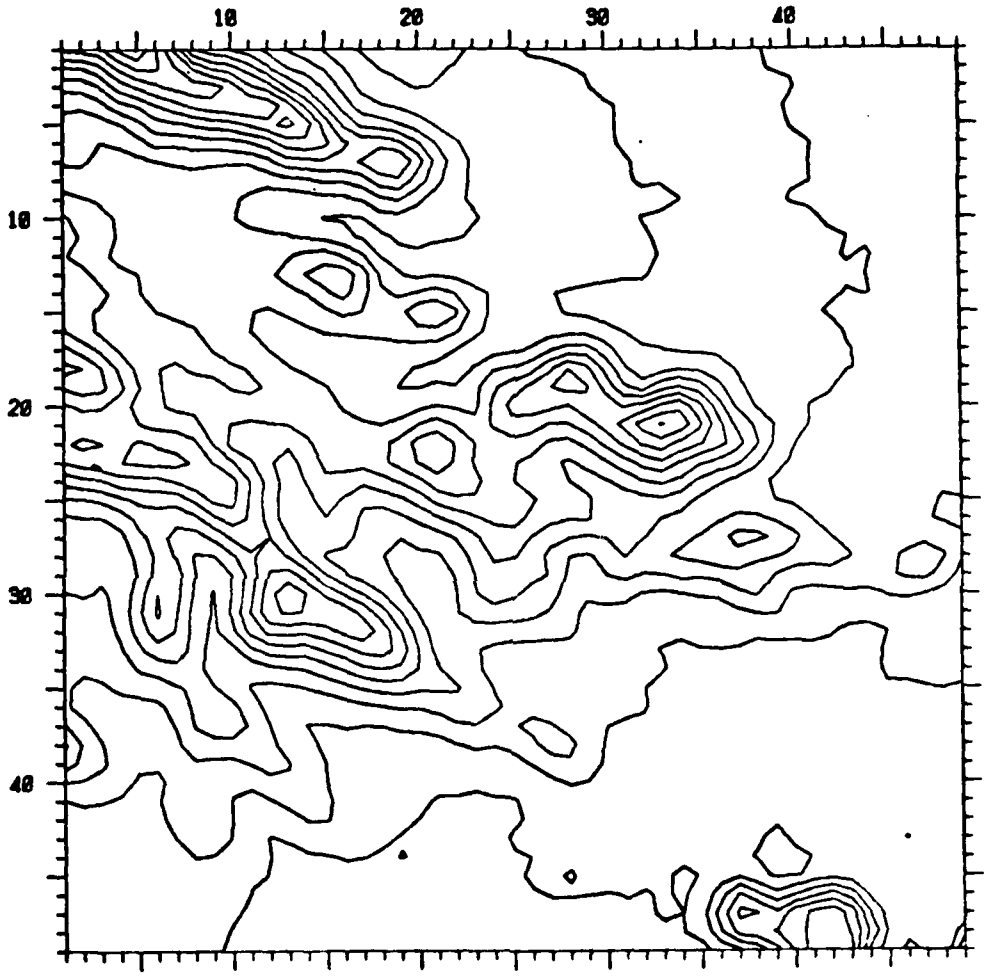


Figure B-3.

This figure shows the data from Figure B-2 after 10 iterations of the algorithm described in Equations (2-5) and (2-6) as modified by Sections B-1 and Equations (B-4) and (B-5). Parameters for the run were DSLTHRESH=.1, DELHEIGHT=1/12, PCNULL=0.0. Contours are at 10 meter intervals.

BIBLIOGRAPHY

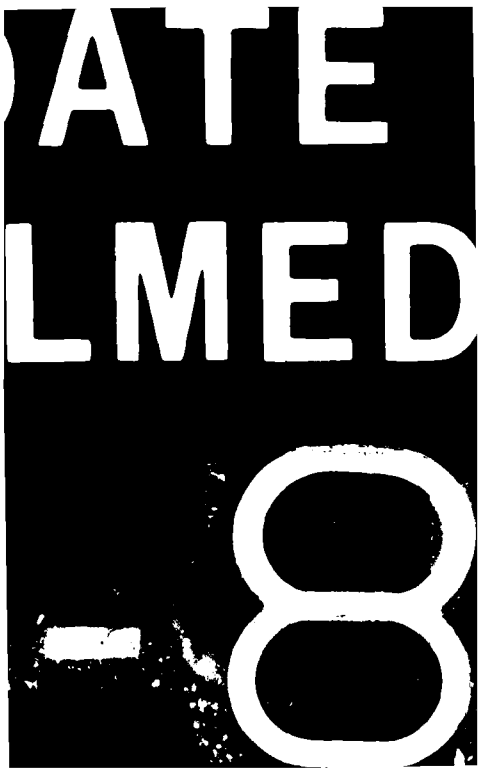
Crombie, Michael A., "Stereo Analysis of a Specific Digital Model Sampled from Aerial Imagery", U. S. Army Engineer Topographic Laboratories Report ETL-0072, September, 1976.

Jancaitis, James R., "Modelling and Contouring Irregular Surfaces Subject to Constraints", U. S. Army Engineer Topographic Laboratories Report ETL-CR-74-19, January, 1975.

Johnson, Charles N., "Processing of Terrain Elevation Data for Improved Contouring", Institute for Advanced Computation Technical Memo 5646, June, 1978.

Quam, Lynn H., "Computer Comparison of Pictures", Stanford University Artificial Intelligence Laboratory Memo AIM-144, July, 1971.

Rosenfeld, Azriel, Robert A. Hummel, and Steven W. Zucker, "Scene Labeling by Relaxation Operations", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-6, No. 6, June, 1976.


ATE
LMED
8